REMARKS

Claims 1-27 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(b) Rejection:

The Examiner rejected claims 1-3, 5-10, 12-14, 16-21, 23, 24 and 27 under 35 U.S.C. § 102(b) as being anticipated by Tredennick et al. (U.S. Patent 4,338,661) (hereinafter "Tredennick"). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations*. The Examiner cited column 15, lines 33-34 and 36-37 as teaching these limitations. These passages describe a micro ROM 72 containing 544 microwords, each having 17 bits, which are addressed by a 10-bit output of address selection block 64. There is nothing in this citation, or elsewhere in Tredennick, that teaches or suggests that these microwords comprise groups of microcode operations, or that a row in Tredennick's microcode ROM stores a plurality of such groups of microcode operations.

Similarly, Tredennick fails to teach or suggest *wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine*. The Examiner cites column 1, lines 55-56 as teaching this limitation. However, this passage discloses only "a data processor having a control store containing a plurality of microinstruction routines for implementing instructions received by the data processor." It says nothing about how these routines are stored within the ROM, or about them comprising a group of microcode operations (from among a plurality of such groups). There is nothing in Tredennick that teaches or suggests this limitation.

Tredennick also fails to teach or suggest *wherein the row stores an associated control sequence for each of the plurality of groups of microcode operations*. The Examiner cites column 15, lines 52-55 and 59-66 as teaching this limitation. These passages describe two formats for microwords. In a microword of a conditional branch type (type II), bits 7 thru 14 comprise a next micro ROM base address (NMBA) for the micro and nano control stores, and are augmented by 2 additional bits supplied by branch control logic (C0 and C1) in order to specify the next address for the control stores (i.e., in order to select a single word line for the micro and nano control stores). In microwords having format type I, bits 2 and 3 comprise a type field (TY) which specifies the source of the next address for the control stores as being from one of the 3 possible addresses provided by the instruction register sequence decoder or from a direct branch address provided by bits 5 thru 14 of the microword. Thus, these passages disclose that information stored in each microword may be used in determining the next address used to select a single word line for each of the control stores. They clearly do not describe a control sequence being associated with a group of instructions comprising a row in the microcode, as in Applicants' claimed invention. There is nothing in these passages or elsewhere that teaches or discloses **a row** storing an associated control sequence for each of the plurality of groups of microcode operations stored in the row.

Further regarding claim 1, Tredennick fails to teach or suggest *a control sequence logic unit coupled to the microcode ROM, wherein in response to accessing the group of microcode operations comprised in the microcode routine, the control sequence logic unit is configured to use the control sequence associated with the group of microcode operations to identify an other row storing one or more next groups of microcode operations comprised in the microcode routine*. The Examiner cited column 15, lines 37-40 as teaching this limitation ("it selects the next line of the ROM, which is output from the microcode ROMs as shown in column 15, lines 52-55 and 59-66.") First, as discussed above, Tredennick does not disclose a control sequence associated with a group of microcode operations. In addition, column 15, lines 37-40 merely describes that the micro ROM is addressed by the 10-bit output of address selection block 64. Furthermore, the Examiner's statement "it *selects the next line of the ROM*" has no basis in Tredennick.

As discussed above, column 15, lines 52-55 and 59-66 describes how the next address to be decoded for the control stores is determined for microwords having type I and type II formats. None of these descriptions includes identifying an other row storing one or more next groups of microcode operations comprised in the microcode routine. Tredennick says nothing about the next address identifying another row, about another row storing one or more groups of microcode operations, or about another row storing one or more next groups of microcode operations comprised in the (same) microcode routine, as in Applicants' claimed invention.

In the Response to Arguments section of the Final Action, the Examiner states, "For further clarification of a row in the ROM containing multiple "groups", see Column 19, lines 23-27, which had been previously referred to in claims such as Claim 6." The Examiner cited this passage in the previous Office Action, submitting, "for each row, the address may represent one, two, four, or up to eight different groups. So there are segments in the sense that some lines can contain a different number of groups than the other lines." However, as noted in Applicants' previous response, **the Examiner has misquoted this passage**. The Examiner's citation actually states, "Each word line in the micro ROM is represented by only one input address. Each word line in the nano ROM however may represent one, two, or four possible different input addresses. In the preferred embodiment of the data processor, a word line in the nano ROM may represent as many as eight different input addresses."

Applicants assert that it is clear from FIG. 9 and its corresponding description above that "a word line" in Tredennick is not a row in the microcode ROM that stores a plurality of groups of microcode operations, as in Applicants' claimed invention. It is not even a row of microcode words (which the Examiner has, in some of his arguments, equated to "groups of microcode operations"). Instead, as noted in Applicants' previous Response, "a word line" is part of the decoding circuitry illustrated in FIG. 9, that selects a single microword and a single nanoword from the micro ROM and nano ROM, respectively, based on an address inputs A_0, A_1, and A_2 on signals 128, 129, and 130, and in the case of conditional branches, on additional signals C0 and C1. (See, e.g., column

19, lines 13-22: "the same address is presented to the decoders of both the micro ROM and the nano ROM. For any input address, there will be no more than one word line in each ROM which remains high. The line which remains high will cause the appropriate output value to be generated as the micro ROM output word and the nano ROM output word according to the coding at the intersection of the selected word line and the output columns.") Furthermore, the Examiner has clearly misinterpreted the cited passage in column 19. Tredennick does not teach one address (or row) mapped to (or containing) multiple groups of microcode instructions, as the Examiner suggests in this remark, "some lines can contain a different number of groups than the other lines." Instead, Tredennick teaches that each individual microword or nanoword may be selected using multiple input addresses (e.g., the decoding circuitry may map many addresses to one word line, not the other way around.)

In the Response to Arguments section, the Examiner also submits, "Column 1, Lines 55-56 show that these instructions are part of a microcode routine, and Column 15, Lines 52-55 and 59-66 show that each word (group) has a control which specifies the address to the next address, which must identify a row in order to continue to use the ROM." While the Examiner's citation in column 1 does describe a control store containing a plurality of microinstruction routines, Applicants again assert that there is nothing in Tredennick that describes that these microinstruction routines are organized in the control store in the specific manner required by Applicants' claims (e.g., *wherein a row in the microcode ROM stores a plurality of groups of microcode operations* and *wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine*, etc.) Furthermore, the Examiner's apparent interpretation that "each word" is equivalent to a "group of microcode operations" has absolutely no basis in the cited art. In addition, the Examiner's suggestion that because each word "has a control which specifies the address to the next address" this "must identify a row in order to continue to use the ROM" is not correct and is completely unsupported in the cited art. The "row" of Applicants' claimed invention is specifically defined in claim 1 as recited above (i.e., it stores a plurality of groups of microcode operations, etc.). The "address to the next address" in Tredennick is a collection of address bits (whose source is bits 5-14

of the microword) that are inputs to address selector 64. Address selector 64 selects these address bits or one of three other sets of address bits provided by the instruction register sequence decoder dependent on the value of bits 1-3 of the microword (see, e.g., FIG. 5, and column 5, line 66 – column 6, line 13.) The selected address bits are then provided to the microrom and nanorom to select the microword and nanoword corresponding to a specific word line, according to the values of the address bits (as described above regarding address bits A_0, A_1, and A_2 on signals 128, 129, and 130 of FIG. 9), not to identify a row, where a row is defined as in Applicants' claims.

To further illustrate that Tredennick does not teach the limitations of Applicants' claims, please see FIGs. 10 and 11, Appendix A, and the descriptions of how the microwords and nanowords of Tredennick are organized. Appendix A includes a list of microwords, listed by microcode routine. However, these microwords are not stored in rows containing groups comprised within their corresponding microcode routines in the microrom and nanorom, as in Applicants' claimed invention. For example, a routine labeled "ablw1" is listed in Appendix A and includes 3 microwords: ablw1, ablw2, and ablw3. FIG. 10 illustrates the location of each of the microwords that comprise this microcode routine. As shown in FIG. 10B, the address (bits A9-A0) used to access ablw1 is 01 11 10 00 10. As shown in FIG. 10C, the address used to access ablw2 is 10 00 11 00 11. As shown in FIG. 10D, the address used to access ablw3 is 11 01 10 10 11. Thus, FIG. 10 clearly illustrates that the microwords comprising this microcode routine in Tredennick are not grouped and stored together in a same row of the microrom, as required by Applicants' claimed invention. Similarly, the nanowords corresponding to these microwords are not grouped and stored together in a same row. For example, the nanoword corresponding to ablw1 is accessed at address 01 111x xx 10 (see, FIG. 11D), the nanoword corresponding to ablw2 is accessed at address 10 00 11 0x 11 (in FIG. 11D), and the nanoword corresponding to ablw3 is accessed at address 11 01 10 10 11 (in FIG. 11F.) These are all stored in different rows of the nanorom.

To further illustrate what the Examiner's citation in column 19, lines 23-27, please also refer to column 19, lines 43 – 55. This passage describes an example of how

microwords and nanowords are mapped, and describes that one nanoword may be addressed by more than one input address. This passage includes the following:

> As an example, assume that the current micro control store address (A9-A0) is the 10-bit code 01 11 10 00 10. This address references the location labeled ablw1 in the micro ROM as is shown in FIG. 10B. This same address references the location labeled ablw1 in the nano ROM as shown in FIG. 11D. As is indicated in the column labeled ORIGIN in Appendix A, other microwords which refer to the same nanoword location include abll1, ralw1, rall1, jsal1, jmal1, paal1, and unlk2.

This passage describes an address accessing the same microword as that described above, ablw1. This passage also describes that several other addresses may be used to access the same nanoword, including, for example, the address used to access microword abll1. Referring to Appendix A and FIGs. 10 and 11, the address used to access microword abll1 is 01 11 10 01 10 (see FIG. 10B.) This address also accesses nanoword ablw1, because the address matches the pattern indicated for nanoword ablw1: 01 11 1x xx 10 (see FIG. 11D). In this example, microword abll1 includes microwords abll1, abll2, and abll3. Microword abll2 is accessed using the address 10 00 11 01 11 (see FIG. 10C). This address also accesses nanoword ablw2, since its address matches the pattern indicated for this nanoword: 10 00 11 0x 11 (see FIG. 11D). Microword abll3 is accessed using address 10 00 11 00 01 (see FIG. 10C) and corresponding nanoword abll3 is accessed using address 10 00 11 0x 01 (see FIG. 11D). In addition, these microwords and nanowords corresponding to microcode routine abll1 are clearly not grouped and stored together in a same row.

Applicants again assert that, as clearly shown in the above examples, **the microrom and nanorom of Tredennick are clearly not organized in the manner required by Applicants' claim 1**, which recites, "wherein a row in the microcode ROM stores a plurality of groups of microcode operations, wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine."

For at least the reasons above, Applicants again assert that Tredennick clearly does not teach all the limitations of Applicants' claim 1. Therefore, the rejection of claim 1 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 12 and 23 include limitations similar to claim 1, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 27, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *a microcode ROM, wherein a row in the microcode ROM stores a plurality of groups of microcode operations and wherein the row stores an associated control sequence for each of the plurality of groups*. In the previous Office Action, the Examiner again cited column 15, lines 33-34, 36-37, 52-55, and 59-66 as teaching these limitations. However, as discussed above regarding claim 1, Tredennick does not teach or suggest a row in the microcode storing groups of microcode operations and a control sequence associated with each of the groups.

Further regarding claim 27, Tredennick fails to teach or suggest *means for accessing a control sequence associated with one of the plurality of groups of microcode operations and responsively accessing a next group of microcode operations stored in the microcode ROM*. The Examiner again cited "Column 15, lines 37-40, it selects the next line of the ROM, which is output from the microcode ROMs as shown in column 15, lines 52-55 and 59-66." However, as discussed above regarding claim 1, Tredennick does not teach or suggest groups of microcode operations stored in a row or a control sequence associated with a group of microcode operations, and therefore, does not teach or suggest accessing such a sequence or responsively accessing a next group of microcode operations stored in the microcode ROM. Therefore, Tredennick cannot be said to anticipate claim 27.

In the Response to Arguments section of the Final Action, the Examiner submits that claim 27, "has similar limitations, except the control addresses a group instead of a row, which is taught by Tredennick as explained above." Applicants assert that, as

discussed above regarding claim 1 and claim 27, Tredennick does not teach a control sequence associated with each one of a plurality of groups of microcode operations, nor that such control sequences are stored in a row containing each of the plurality of groups of microcode operations, as required by claim 27.

Furthermore, Applicants note that the limitations of claim 27 are not similar to those of claim 25, "except the control addresses a group instead of a row", as suggested by the Examiner. For example, claim 1 does not recite accessing a next row of microcode operations in response to accessing the control sequence, but instead recites identifying an other row storing one or more next groups of microcode operations comprised in the microcode routine using the control sequence associated with the group of microcode operations. Applicants note that these differences were not addressed in the Examiner's Response to Arguments, and assert that they are not taught by Tredennick.

For at least the reasons above, the rejection of claim 27 is unsupported by the cited art and removal thereof is respectfully requested.

Regarding claim 2, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *at least one of the plurality of groups of microcode operations stored in the row is part of a different microcode routine*. The Examiner cited Figure 11 and column 19, lines 52-55 as teaching this limitation. As discussed above, this passage and figure illustrate how only one input address may reference a particular (single) microword location but one, two, or four input addresses may reference the same (single) nanoword location in order to reduce the size of the control store (see, e.g., column 19, lines 13-34.) This has nothing to do with whether one of a plurality of groups of microcode operations stored in a row is part of a different microcode routine. As discussed above regarding claim 1, Tredennick does not teach or suggest a microcode ROM organized according to the rows, routines, and groups of Applicants' claimed invention. Instead, individual microwords and nanowords are stored in the rows of Tredennick. They are not grouped according to the microcode routines in which they are included.

In the Response to Arguments section of the Final Action, the Examiner submits, "Figure 11 clearly shows that each row contains multiple instructions of different types, thus different groups. As group has been given no solid definition in the claims, different groups have been interpreted as different instructions. Figure 11 also for this reason shows each line containing multiple groups per row." Applicants disagree that the claims do not provide any definition of what a "group" is, or that instructions of different types may be equated to the "groups" of Applicants' claimed invention. For example, Applicants' claim 1 clearly recites, "*wherein a group of the plurality of groups of microcode operations is comprised in a microcode routine*." The Examiner's broad interpretation, "different groups have been interpreted as different instructions" is clearly inconsistent with this limitation recited in claim 1. Furthermore, as discussed above, **the rows in Tredennick do not include such groups of microcode operations comprised in a microcode routine** or multiple groups, one of which may be comprised in a different microcode routine, as required by this claim.

For at least the reasons above, the rejection of claim 2 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 13 includes limitations similar to claim 2, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 3, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *wherein the control sequence logic unit is configured to identify which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine based on information contained in the control sequence associated with the group of microcode operations stored in the row*. The Examiner cited column 15, lines 52-55 and 59-66, "which defines which row and position the next group is located," as teaching these limitations. However, as discussed above, this citation does not teach identifying a row and position for a next group of microcode operations. Instead, this passage describes how the next address (which is clearly not the same as "a row and position") is determined for an individual microword

having a type I or type II format. As discussed above regarding claim 1, Tredennick does not teach or suggest a microcode ROM organized according to the rows, routines, and groups of Applicants' claimed invention. Furthermore, there is nothing in the Examiner's citation or elsewhere in Tredennick that teaches or suggests that identifying the next address (or "a row and position") has anything to do with identifying "which of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine," as recited in claim 3. As discussed above regarding claim 1, **the microwords and nanowords of Tredennick are not grouped by microcode routine**, and thus may not be used to identify a group of microcode operations stored (together) in another row of the microcode ROM, as required by this claim.

In the Response to Arguments section of the Final Action, the Examiner disagrees with Applicants' argument that an address does not specify a row and position for the next group. The Examiner submits, "The cited portion discloses that the address directs the processor to the next group in the routine, which must specify a row, as discussed in Claim 1. It can be further seen in Figure 11a that the 10-bit address does select a position using bits A1 and A0." Applicants again assert that, as discussed above, the address selects a single microword or nanoword, not a group of microcode operations stored in a row and comprised in the microcode routine, as **Tredennick does not teach this grouping of microcode operations comprising a microcode routine stored together within a row (or stored together in more than one row) of the microrom or nanorom.**

For at least the reasons above, the rejection of claim 3 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 14 includes limitations similar to claim 3, and so the arguments presented above apply with equal force to this claim, as well.

Regarding claim 5, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *wherein if the group of microcode operations comprises at least one branch operation, the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine dependent on a branch prediction as well as the control sequence associated with the group of microcode operations*. The Examiner cited column 15, lines 49-55 ("teach a microcode instruction for branches") and column 17, lines 29-32 ("show that the outcome either way will be in the same row specified by the control sequence") as teaching these limitations. The Examiner's citation in column 17 states, "Thus, two microwords which serve as alternate destinations for a particular conditional branch type microword must be placed in the same logical row of the micro ROM." Applicants assert that this citation does not describe identifying the next group of microcode operations in a microcode routine, but instead describes two alternate (individual) addresses that may be the destination of a branch operation. Furthermore, the Examiner's remarks appear to teach away from identifying the next group of operations dependent on a branch prediction. Instead, they imply that no branch prediction is necessary (or performed) because "the outcome either way will be in the same row." In fact, branch prediction is not disclosed in Tredennick. Finally, Applicants' claim 5 does not recite identifying a row to be accessed after a conditional branch, as the Examiner suggests, but instead recites, "the control sequence logic unit is configured to identify the next group of microcode operations in the microcode routine…" which Tredennick clearly does not disclose. Therefore, Tredennick cannot be said to anticipate claim 5.

In the Response to Arguments section of the Final Action, the Examiner submits, "Tredennick teaches that the branch outcomes must be in the same row of the microcode ROM (further providing evidence of multiple groups per ROM row, which Applicant asserted was not taught in the reference.)" Applicants again assert that this teaches only that multiple individual microwords may be contained in a row, not **multiple groups of microcode operations having the limitations recited in Applicants' claims**. The Examiner further submits, "Applicant has argued that Tredennick teaches away from the claims because both outcomes are located in the same row. However, Tredennick

teaches this is done to minimize the size of storage, to prevent multiple instances of the branch from being put in memory (Column 2, Lines 40-50), and as explained above, Tredennick can specific positions inside a row, the fact that they are in the same row is space optimization."

The Examiner has misunderstood Applicants' argument. Applicants' argument was that Tredennick taught away from identifying a row containing a next group of instructions to be accessed by a conditional branch based on a branch prediction for several reasons. One is that all of the possible next microwords to be accessed will be in the same row, therefore the next row to be accessed is already known. The fact that the outcomes are positioned in the same row to save space is irrelevant. The second is that no branch prediction is described, as required by Applicants' claim. In addition, Tredennick does not teach identifying a row containing a next group of microcode operations comprised in the microcode, as required by Applicants' claim, only a next microword (and/or nanoword). As discussed above, Tredennick does not teach that microcode operations are so grouped.

For at least the reasons above, the rejection of claim 5 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 16 and 24 include limitations similar to claim 5, and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 6, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *wherein the microcode ROM is divided into a plurality of segments, wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments*. The Examiner cited column 19, lines 22-27 as teaching this limitation ("for each row, the address may represent one, two, four, or up to eight different groups. So there are segments in the sense that some lines can contain a

different number of groups than the other lines.") As noted above, the Examiner has misquoted column 19. The Examiner's citation states, "Each word line in the micro ROM is represented by only one input address. Each word line in the nano ROM however may represent one, two, or four possible different input addresses. In the preferred embodiment of the data processor, a word line in the nano ROM may represent as many as eight different input addresses." However, "a word line" in Tredennick is not a row in the microcode ROM that stores a plurality of groups of microcode operations, as in Applicants' claimed invention. Instead, "a word line" selects a single microword and/or a single nanoword from the micro ROM and nano ROM, respectively. (See, e.g., column 19, lines 13-22: "the same address is presented to the decoders of both the micro ROM and the nano ROM. For any input address, there will be no more than one word line in each ROM which remains high. The line which remains high will cause the appropriate output value to be generated as the micro ROM output word and the nano ROM output word according to the coding at the intersection of the selected word line and the output columns." There is nothing in Tredennick that teaches or suggest the microcode ROM is divided into a plurality of segments having the limitations recited in claim 6 ("wherein a same number of groups of microcode operations is stored in each row of a given one of the plurality of segments, and wherein each row in the given one of the plurality of segments stores a different number of groups of microcode operations than each row in each other one of the plurality of segments.") Therefore, Tredennick cannot be said to anticipate claim 6.

In the Response to Arguments section of the Final Action, the Examiner submits that "in Tredennick, the word line in one ROM is a row (the micro ROM), or a "group" in the nano ROM. Thus, when reading Tredennick as it was intended, a "row" or "word" in the micro ROM corresponds to multiple "words" in the nano ROM." **The Examiner is incorrect**. As discussed above, Tredennick teaches that one input address may be used to identify one word (which is not a "row" as defined by Applicants' claims) in the micro ROM and one word in the nano ROM. The mapping of multiple nano words to one micro word described by the Examiner is incorrect. Instead Tredennick teaches that more

than one input address may identify the same nano word, as in the example detailed above.

The Examiner further submits, "Tredennick further teaches, in Column 19, lines 23-27, that the nano ROM may represent one, two, four, or up to eight different groups, and that each line may do so, which the Examiner had interpreted such that each line may have one, two, four, or eight associated groups/operations, such that "segments" were formed, based on the number of groups in each corresponding micro ROM row." However, as discussed at length above regarding claim 1, the Examiner has both misquoted and misinterpreted this passage of Tredennick, which does not teach the grouping of microcode operations defined by Applicants' claims.

For at least the reasons above, the rejection of claim 6 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 17 includes limitations similar to claim 6, and so the arguments presented above apply with equal force to this claim, as well.

Similarly, Tredennick fails to teach or suggest the limitations of claims 7-9, which recite, in part, *wherein groups of microcode operations stored in a same one of the plurality of segments have a same maximum width* (claim 7), *wherein groups of microcode operations stored in one of the plurality of segments have a maximum width that is different from a maximum width of groups of microcode operations stored in another one of the plurality of segments* (claim 8), and *wherein one of the plurality of segments stores one group of microcode operations and one associated control sequence per row* (claim 9.) The Examiner cited column 19, lines 22-27 as teaching all of these limitations. However, as discussed above, this citation has nothing to do with a plurality of segments in a microcode ROM, much less with such segments having the limitations recited in claims 7-9.

In the Response to Arguments section of the Final Action, the Examiner states that by the same reasoning applied to his rejection of claim 6, Tredennick reads on claims 7-9 and 18-20. However, as discussed above regarding claims 1 and 6, Tredennick clearly does not teach anything about the groups or segments of microcode operations defined or stored as in Applicants' claimed invention. Furthermore, Applicants again assert that Tredennick teaches nothing about the maximum width of any such segments or what would be stored in such a segment (e.g., *one group of microcode operations and one associated control sequence per row*, as claimed.) It is clear from the detailed examples above of how microwords and nanowords are stored in Tredennick, none of these limitations are taught by Tredennick. Therefore, Tredennick cannot be said to anticipate these claims.

For at least the reasons above, the rejection of claims 7-9 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 18-20 include limitations similar to claims 7-9 and so the arguments presented above apply with equal force to these claims, as well.

Regarding claim 10, contrary to the Examiner's assertion, Tredennick fails to teach or suggest *wherein the control sequence logic unit is configured to identify a position of one or more groups of microcode operations and a position of one or more control sequences dependent on which of the plurality of segments of the microcode ROM stores the one or more groups of microcode operations*. The Examiner cited column 15, lines 52-55 and 59-66, "which defines which row and position the next group is located," as teaching this limitation. However, as discussed above, this citation this citation does not teach identifying a row and position for a next group of microcode operations. Instead, this passage describes how the next address (which is clearly not the same as "a row and position") is determined for an individual microword having a type I or type II format. As discussed above, Tredennick does not teach or suggest a microcode ROM organized according to the rows, routines, groups, and segments of Applicants' claimed invention. Therefore, Tredennick clearly cannot (and does not) teach identifying a

position of one or more groups of microcode operations and control sequences dependent on which of a plurality of segments stores the one or more groups of microcode operations. Therefore, Tredennick does not anticipate claim 10.

In the Response to Arguments section of the Final Action, the Examiner submits, "the arguments made are substantially similar to the arguments for Claims 3 and 14, and Examiner refers Applicant to the arguments for those Claims." Applicants assert that, as discussed above, Tredennick **does not teach the limitations of claims 3 and 14**, nor does Tredennick teach the additional limitations of claim 10 (e.g., those directed toward a plurality of segments and which one stores the one or more groups of microcode operations.)

For at least the reasons above, the rejection of claim 10 is unsupported by the cited art and removal thereof is respectfully requested.

Claim 21 includes limitations similar to claim 10, and so the arguments presented above apply with equal force to this claim, as well.

Section 103(a) Rejection:

The Examiner rejected claims 4, 11, 15, 22 and 25-26 under 35 U.S.C. § 103(a) as being unpatentable over Tredennick in view of Yoshida (U.S. Patent 5,761,470). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 4, contrary to the Examiner's assertion, Tredennick in view of Yoshida fails to teach or suggest *wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler.* The Examiner submitted that

Tredennick teaches the microprocessor of claim 3, which Applicant traverses above. The Examiner admits that Tredennick fails to teach the above-referenced limitation of claim 4, and relies on Yoshida to teach it. Yoshida teaches of a VLIW machine, which exploits parallelism by executing multiple instructions simultaneously. Yoshida teaches that if the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place, as it has to execute some instruction (column 1, lines 51-56). The Examiner submits that, "Given the advantage of higher speed though parallelism, one of ordinary skill in the art at the time the invention was made would have converted Tredennick's invention to operate in a parallel fashion such as a VLIW machine to increase the speed and performance."

Applicants remind the Examiner that, as stated in the MPEP §2143.01 "If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti* , 270 F.2d 810, 123 USPQ 349 (CCPA 1959). . ." (*emphasis added*). Applicants assert that converting Tredennick's invention to operate in a parallel fashion would clearly (and dramatically) change the principle of operation of his invention.

Applicants also remind the Examiner that to establish a *prima facie* obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. Applicants assert that converting Tredennick's invention "to operate in parallel fashion" would not necessarily result in Applicants' claimed invention. The "feature" described in Yoshida "if the conventional VLIW machine cannot execute an instruction from the word in parallel, it inserts a NOP in its place" refers to a VLIW processor that consumes one instruction even when there are no operations which can be executed in parallel. In this case, a number of operation fields specifying null operations (No Operation: NOP) are generated, and the amount of instruction code becomes very big.

In addition, the cited feature of Yoshida is clearly not the same as the limitation in Applicants' claim 4, which recites "*wherein if fewer than all of a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine, the control sequence logic unit is configured to substitute NOPs for the microcode operations comprised in the groups not comprised in the microcode routine when outputting the row to the scheduler,*" as Yoshida does not describe inserting NOPs for groups of operations as in Applicants' claimed invention. Furthermore, it is not clear that all parallel processors necessarily include this "feature" of Yoshida, nor does the reference (or the Examiner, in his remarks) explain how this feature would be implemented in Tredennick's processor if "converted to operate in parallel fashion." **In fact, this "feature" is described as a deficiency of the prior art, rather than a feature of the invention of Yoshida that one would be motivated to include in Tredennick.**

In the Response to Arguments section of the Final Action, the Examiner disagrees with Applicants' argument that the combination of Yoshida would clearly and dramatically change the principle of operation of this invention. The Examiner submits that Tredennick's invention is essentially a machine which executes instructions using microcode stored in a ROM and that modifying the invention to work in a parallel manner does not affect this principle in any way, shape, or form. Applicants again assert, however, that modifying the machine of Tredennick "to operate in a parallel fashion" in order to obtain an "advantage of higher speed through parallelism" would not necessarily result in Applicants' claimed invention. There are numerous ways in which a parallel processor may schedule instructions, not all of which **necessarily require** the referenced feature of Yoshida. It is clear that "higher speed through parallelism" may be achieved without this feature. In fact, it appears that this feature may in some instances actually limit the speed of operation, as it requires the machine to consume one instruction (and one instruction cycle) even when there is no operation that can be executed in parallel. Therefore, Applicants again assert that the Examiner has not provided a proper motivation to combine the cited references.

The Examiner further submits, in the Response to Arguments section, that "Yoshida is used to show that one of ordinary skill in the art would have recognized the need to insert NOP's in place of instructions fed to the machine that could not be executed in order to ensure correct execution, which is a motivation for doing so to one of ordinary skill in the art." However, the Examiner's stated "need" does not exist in Tredennick, nor is it clear that it exists in all machines that operate in a parallel fashion. There is also nothing in Yoshida that describes that this feature is needed to ensure correct operation of a parallel machine, as suggested by the Examiner. In fact, as noted above, Yoshida itself teaches that the number of such added NOP operations should be reduced in order to improve code efficiency (see, e.g., column 1, line 67 – column 2, line 7.) Therefore, one would not be motivated to add this feature to Tredennick, as it would not necessarily contribute to improved speed (even if Tredennick were converted to operate in parallel) and it would reduce code efficiency, as taught by Yoshida.

In addition, Applicants again assert that even if properly combined, the result does not teach the limitations of claim 4, which recites limitations concerning substituting NOPs for the microcode operations comprised in **groups** not comprised in the microcode routine (rather than adding NOP instructions in individual empty slots in a scheduler.)

For at least the reasons above, the rejection of claim 4 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 15 and 25 include limitations similar to claim 4, and so the arguments presented above apply with equal force to these claims, as well.

Similarly, Tredennick in view of Yoshida fails to teach or suggest *wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access*, as recited in claim 11. The Examiner admitted that Tredennick fails to teach this limitation, and again relies on Yoshida's VLIW machine to teach this limitation. As discussed above, the Examiner's proposed modification of the prior art would change the principle

of operation of the prior art invention being modified, and thus the teachings of the references are not sufficient to render the claims *prima facie* obvious.

In addition, the Examiner's citations in Yoshida do not teach *a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in the microcode routine and are output during a single access*. They teach that one VLIW word can specify a plurality of instructions (column 1, lines, 25-30), but they do not disclose that instructions are output during a single access, that they are stored in rows, or wherein a plurality of groups of microcode operations stored in the other row of the microcode ROM are comprised in a microcode routine.

In the Response to Arguments section, the Examiner states that "Yoshida does not need to teach instructions stored in rows in a ROM, and the other structural elements of this claim, that is what Tredennick was used to teach." However, as discussed above regarding numerous other claims, **Tredennick does not teach the structural elements of the microcode ROM recited in Applicants' claims**, such as the rows, groups and segments of microcode operations comprised in microcode routines. The Examiner submits, "Yoshida provides motivation for accessing multiple groups of the groups in the row that Tredennick teaches, to increase performance by being able to potentially execute multiple operations at once, and that one of ordinary skill would have been able to modify Tredennick to do so." However, since Tredennick does not teach these multiple groups of groups in the row, even if references were properly combined, the result would not teach Applicants' claimed invention.

For at least the reasons above, the rejection of claim 11 is unsupported by the cited art and removal thereof is respectfully requested.

Claims 22 and 26 include limitations similar to claim 11, and so the arguments presented above apply with equal force to these claims, as well.

CONCLUSION

Applicants submit the application is in condition for allowance, and notice to that effect is respectfully requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5500-91700/RCK.

Also enclosed herewith are the following items:

☐ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: December 4, 2006